Extra Credit 2

CMSY-199, Spring 2013

The source code for this assignment must be submitted electronically using the Canvas course website prior to the start of class on Monday, May 6.

1. Write a subclass of the `Calculator` class called `ScientifcCalculator` which has all the functionality of its superclass and the additional features described below.

2. Make the `ScientificCalculator` class a Java application by adding a `main` method. Call the `setTitle` method of with the argument `"Scientific Calculator"` and call the `setSize` method to make the application 600 by 300 pixels.

3. Write a method called `changeColors` which embellishes the user interface by changing the background colors and font color. You will need to change the access modifier of some members in the `Calculator` class to allow the `ScientificCalculator` to inherit them. Change the color of the clear to button blue, the equals button to orange, the numeric buttons (including the decimal point and plus minus buttons) to light gray, and all other buttons to dark gray. Change the background of the display to black and change the color of the font on all the above components to white.

4. In addition to the member variables that are inherited by the `ScientificCalculator`, you will need a no-argument constructor and a new version of the `makeButton` method. You will also need to create addtional member variables:

   (a) A `JPanel` container to hold the original buttons.

   (b) A `JPanel` container to hold new buttons (see below).

   (c) A `JPanel` container to hold the other two `JPanel` objects.

   (d) Fifteen `JButton` objects for the scientific functions factorial, square root, percent, cosine, tangent, natual logarithm, base 10 logartihm, reciprocal, natural exponential, square, raise to power, absolute value, and the mathematical constants e and π.

5. Change the icon of the application from the default Java icon to the image in the file `calc.png` shown below.

6. Since the superclass `Calculator` implements the `ActionListener` interface, the `ScientificCalculator` also implements `ActionListenter` whether or not you explicitly say so in the subclass declaration. Override the `actionPerformed` method to provide functionality for the new buttons - the very first line should call the superclass verison of `actionPerformed` and then provide event-handling for the new buttons. The event-handling routines for many of the new buttons can simply delgate to a method in the `Math` class.

7. After you have completed the `ScientificCalculator` class, make sure that the `Calculator` class still functions properly as a standalone application.